LAB REPORT

Abgabe 4

David Goricanec

Lucas Pallan

Repository: https://github.com/LucasPallan/19-Abgabe04-Goricanec-Pallan

VORGEHEN

TASKLISTE

- [X] Taskliste in Readme einfügen

- [X] Queue Vorlage in das Repository einspielen

- [X] Korrigieren des Source Codes

- [X] Erstellung Klassen- und Methodenkommentare mittels Javadoc

- [X] Erstellung Junit Tests

- [X] Anpassung pom.xml

- [X] Log4j integrieren

- [ ] Maven Size Dokumentation erstellen

- [X] Erstellung Markdown Lab Report

- [X] Vollständigkeit der Abgabe überprüfen

- [X] Abgabe PDF Version

VORBEREITUNG

Treffen beim Lucas, Repository erstellen, Dokumentation vorbereiten, Angabe sorgfälltig durchlesen und Aufteilung der Arbeit.

BUGFIXING IM CODE

1. String poll() Zeile 34; "== 0"-Abfrage muss auf "!= 0" geändert werden ("> 0" funktioniert auch)

2. String remove(); element bekommt einen Wert von der Poll Methode und wird sofort mit "" überschrieben -> "element = "";" entfernen

3. Konstruktor Übergabeparameter maxsize -> S wird großgeschrieben

```java
package at.fhj.iit;

import java.util.ArrayList;
import java.util.List;
import java.util.NoSuchElementException;

// there's some Bugs included, try to debug the code and fix the Bugs
// there are different Bugs, wrong implementation, typos, ...
// write Test-Cases (read Queue Interface for understanding methods) and use Debugging possibilies

public class StringQueue implements Queue {

    private List<String> elements = new ArrayList<String>();
    private int maxSize = 5;

    public StringQueue(int maxSize){
        maxSize = maxSize;
    }

    @Override
    public boolean offer(String obj) {
        if(elements.size()!= maxSize)
            elements.add(obj);
        else
            return false;

        return true;
    }

    @Override
    public String poll() {
        String element = peek();

        if(elements.size() != 0){
            elements.remove(0);
        }

        return element;
    }

    @Override
    public String remove() {
        String element = poll();
        if(element == null)
            throw new NoSuchElementException("there's no element any more");

        return element;
    }
```

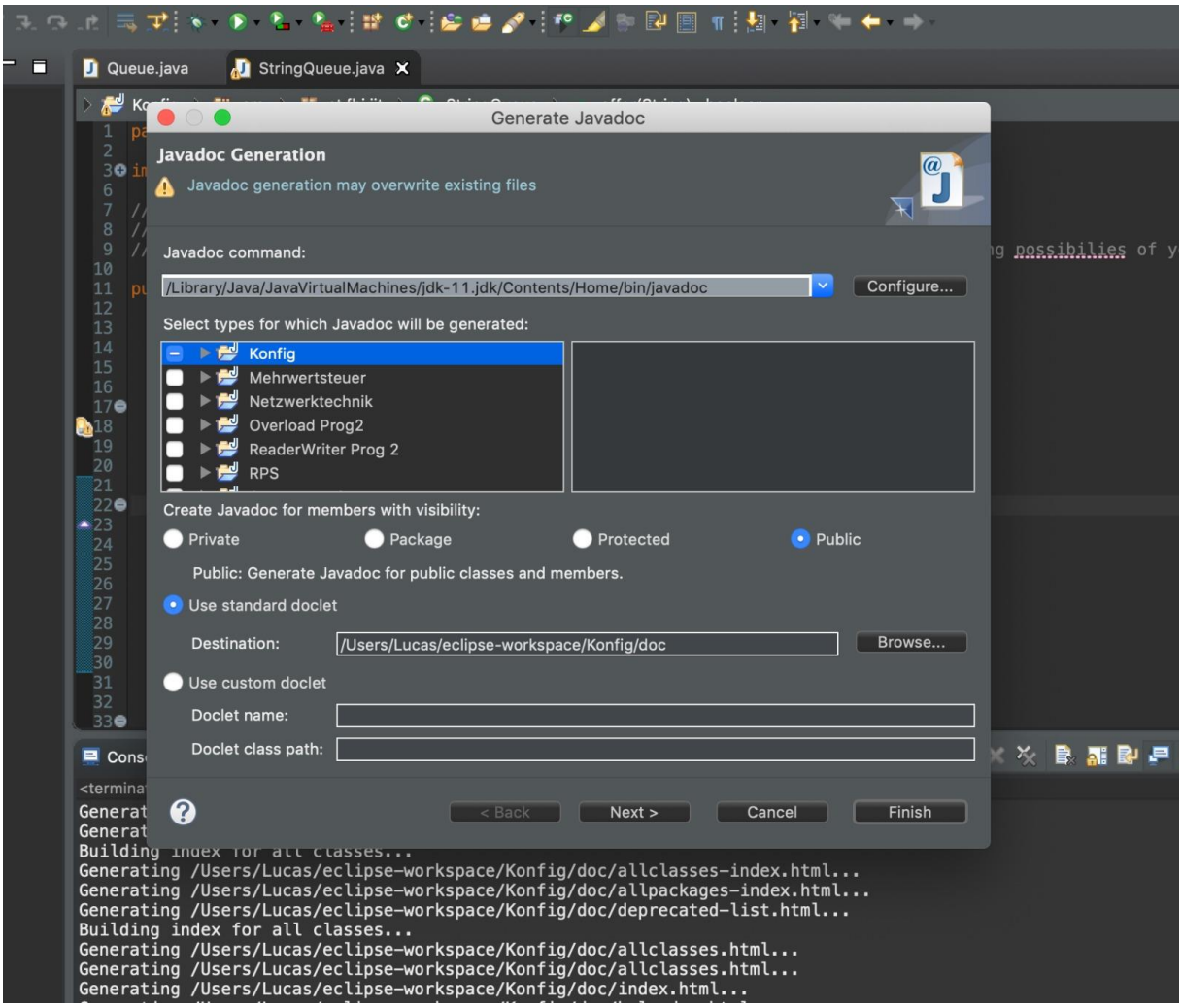
JAVADOC KOMMENTARE

Mittels /** **/ bzw: /** */ werden Kommentare vor jeder Methode erstellt.

Wir versuchen gute, kurze, aussagekräftige Kommentare zu erstellen.

In Ecliipse gehen wir auf "Project Generate Java-Doc" um die Dokumentation zu generieren.

Dann befindet sich die Doku im "Doc"-Order



JavaDoc Result:

Constructors

Constructor	Description
`StringQueue(int maxSize)`	Constructor

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
java.lang.String	`element()`	This method returns the first element.
boolean	`offer(java.lang.String obj)`	This method adds the "obj" element to the queue
java.lang.String	`peek()`	This method returns the first element
java.lang.String	`poll()`	This method returns the first element and deletes it
java.lang.String	`remove()`	This element returns the first element and deletes it.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

JUINT TESTS

Erstellt wurden Test für jede Methode, wenn es Daten und und wenn sie keine Daten hat.

Dabei wird jeder Schritt überprüft.

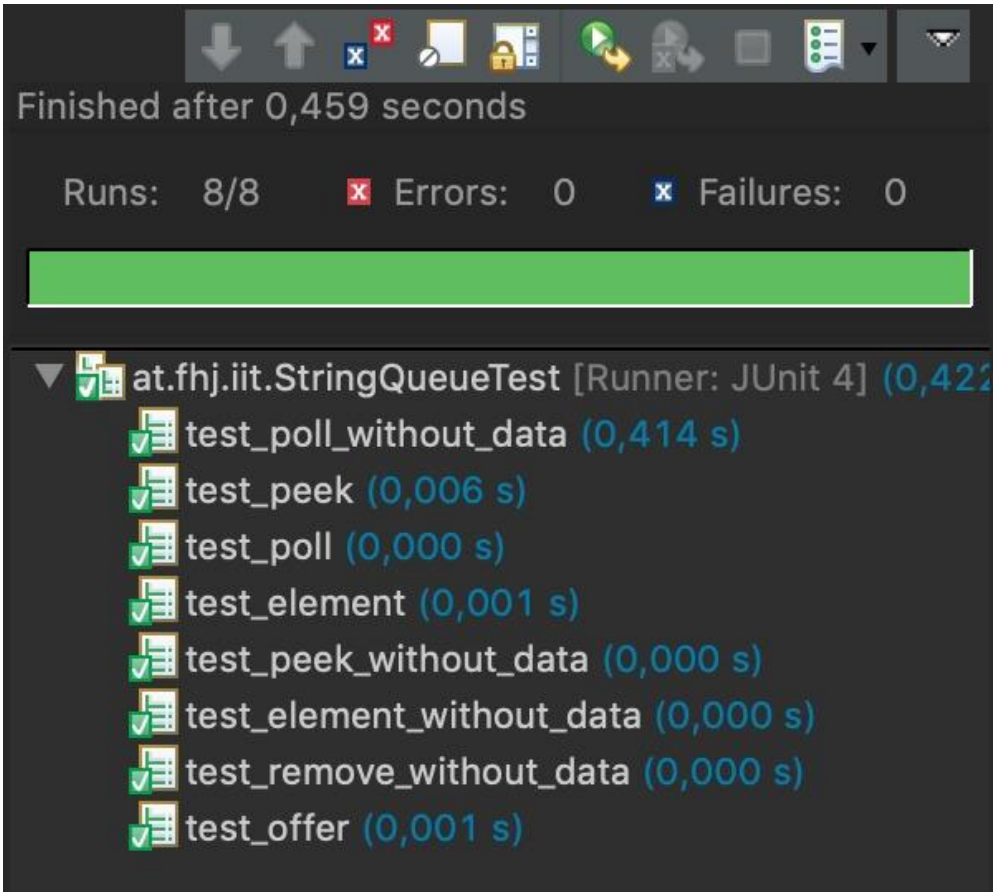
@Before wird ausgeführt, vor dem test und @After nach dem Test

```java
/** JUnit Tests of the StringQueue */
public class StringQueueTest {
    private StringQueue test_StringQueue;
    /** before the tests -> create a Queue with a Maxsize of 5 */
    @Before
    public void setup()
    {
        test_StringQueue = new StringQueue(2);
    }

    /**
     * Add Test Objects and see if they are added in the right order and remove them
     */
    @Test
    public void test_offer() {
        Assert.assertTrue(test_StringQueue.offer("OBJ1"));
        Assert.assertTrue(test_StringQueue.offer("OBJ2"));

        Assert.assertEquals("OBJ1", test_StringQueue.remove());
        Assert.assertEquals("OBJ2", test_StringQueue.remove());
    }

    /**
     * Test deleting non-existing Object
     */
    @Test (expected = NoSuchElementException.class)
    public void test_remove_without_data() throws NoSuchElementException {
        test_StringQueue.remove();
    }

    /**
     * Test peek, should return first element
     */
    @Test
    public void test_peek()
    {
        Assert.assertTrue(test_StringQueue.offer("OBJ1"));
        Assert.assertTrue(test_StringQueue.offer("OBJ2"));

        Assert.assertEquals("OBJ1", test_StringQueue.peek());
    }
    /**
     * Test peek, should return first element
     */
```

Aufgeräumt sollte auch werden:

```
    /**
    * Clean up after test
    */
    @After
    public void cleanup()
    {
        try
        {
            while (true)
            {
                test_StringQueue.remove();
            }
        }
        catch (NoSuchElementException e)
        {
            //nothing to do here, everything is deleted
        }
    }
}
```

Result von den JUnit-Tests:



POM.XML

Anpassung der Developer Informationen in der pom.xml.

Plugins für Maven Site und Javadoc wurden hinzugefügt.

LOG4J INTEGRATION

Zuerst wurde ein statischer Logger erstellt welcher in die Konsole schreibt.

Konfiguriert wurde der Logger in der pom.xml

Eine Info Lognachricht wurde jeder Methode hinzugefügt und ggf um eine Error Lognachricht ergänzt

```java
public class StringQueue implements Queue {

    private static final Logger logger = LogManager.getLogger(StringQueue.class);
    private List<String> elements = new ArrayList<String>();
    private int maxSize = 5;

    /** Constructor **/
    public StringQueue(int maxSize){
        logger.info("Calling StringQueue constructor");
        maxSize = maxSize;
    }

    /** This element returns the first element and deletes it. If the queue is empty "NoSuchElementException" will be
    @Override
    public String remove() {

        logger.info("Calling method 'remove'.");

        String element = poll();
        if(element == null)
            throw new NoSuchElementException("there's no element any more");
        logger.error("throw NoSuchElementException");
        return element;
    }
```

PYHTON / JAVA-CODE

GitHub-Flavor: 3 zeilen Python Code:

```
print("Existing is pain")

s = "String"

x = 0
```

In Java:

```
System.out.println("Existing is pain");

String s = "String";

int x = 0;
```

LINKS / LITERATUR

Stack Overflow: https://stackoverflow.com